

Occupancy Grid Map Merging Using Feature Maps

Yasser Alnounou, Mark J. Paulik, Mohan Krishnan, Greg Hudas, James Overholt

Abstract - Multi-robot exploration and mapping studies have demonstrated that it is often more efficient to explore unknown areas in parallel rather than with a single agent. However, the question of how to integrate individual maps into a consistent global map remains an open research area. This problem, known as map merging, comprises the establishment of a frame of reference for multiple mobile robots, the identification of regions of map overlap, and the integration of individual maps to produce a global result. In this work, we build a hybrid map which integrates occupancy grid and feature data to solve this problem. This integrated representation permits fast and effective map merging. Experimental results are presented that demonstrate algorithm performance in a realistic scenario.

Key terms: multi-agent, multi-robot, map merging, hybrid maps, cooperative robotics, data fusion, robot sensing.

I. Introduction

Exploration and mapping have been a subject of interest in robotics research over the last two decades. A single robot can explore an unknown environment and build a map. However, it is often more efficient when the task is distributed between multiple robots, each one exploring a different region [1]. Saving time and energy, multi-agent systems can explore an unknown area cooperatively in applications such as surveillance, search and rescue, and military reconnaissance. For example, experiments conducted by Howard et. al. [2] used a team of autonomous mobile robots to address indoor reconnaissance. The distributed robotic system consisted of 80 heterogeneous robots, each equipped with sensors which enabled the robots to undertake mapping, object detection and navigation. Such multi-agent systems face many challenges including task coordination, goal selection for exploration, mapping, reliable network design and implementation, etc.

The variety and accuracy of a robot's sensor complement affects the pose estimate quality, and thus, a number of probabilistic mapping techniques have been developed to ameliorate the consequences of pose estimate errors [3]. When multi-agent systems are employed to generate environmental maps, it becomes necessary to merge or integrate individual robot maps to produce a consistent global map, and individual and relative robot pose estimates take on further importance. Most of the techniques described in the literature assume knowledge of the relative poses for the robots [4]. However, if the robots do not initially know where the other robots are, the problem becomes more challenging. One interesting solution, is to localize a robot in the other robots' partial maps if possible, as implemented in [5]. Assuming each robot begins exploring at different starting points, once two robots can communicate, they send their odometry data, LIDAR observations, and maps to each other. Each robot then attempts to localize itself in the received map and make a hypothesis for its relative location. The hypothesis is actively verified by requiring that the robots meet at an agreed time and place. Then, they merge the two maps into a single global map and undertake the same procedure with the other robots. Others have attempted to merge maps without depending on odometry or observation information [6], [7]. In this case, having two occupancy grid maps, from single robot or multiple robots, one needs to detect the overlapping areas and merge them into one global map if possible. Some pose the problem as a search for an optimum transformation that overlaps two partial maps [6]. The process of searching is an optimization problem over one rotation and two translation parameters. Thus, the task is to maximize an objective function that measure how well two maps agree. In reality, the search is arbitrary and spread over the transformation space. For example, in [6], the authors adopt an adaptive random walk algorithm for the search process. A heuristic similarity metric guides the process for selecting a direction in the search algorithm. Additionally, an acceptance indicator is used to validate the transformations. Although convergence of the algorithm is proven, it requires substantial computation to obtain the correct transformation. The work was experimentally verified via simulation on small maps, (400*400 matrices) for which every cell, even unknown, is processed. The average required number of steps necessary for algorithm convergence for this size map is about 40,000, where each step requires approximately 4 msec in a normal-speed computer (Pentium IV, 2.2 GHz). As a result, each merging operation could take one to two minutes. Therefore, the approach is best suited for offline applications, or for

-
- Yasser Alnounou, Mark J. Paulik, and Mohan Krishnan are with the Advanced Mobility Laboratory, Department of Electrical and Computer Engineering at the University of Detroit Mercy, 4001 W. McNichols Rd., Detroit MI, 48221, Corresponding Author Email: paulikmj@udmercy.edu
 - Greg Hudas and James Overholt are with U.S. Army RDECOM-TARDEC, Joint Center for Robotics (JCR), RDTA-RS, Mail Stop: 263, Bldg. 200c, 6501 E. 11 Mile Rd., Warren, MI 48397-5000.

This work was undertaken in the Advanced Mobility Laboratory, at the University of Detroit Mercy, and was funded by the US Army Tank Automotive Research Development Engineering Center (TARDEC), Sponsor Ref. No. W56HZV-04-2-0001, Sub. Contract #3001078040. Document edits: G2- x26262, Public Affair's- x28820, IP Law- x25420

Report Documentation Page			Form Approved OMB No. 0704-0188		
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 01 NOV 2010		2. REPORT TYPE Journal Article		3. DATES COVERED 01-11-2010 to 01-11-2010	
4. TITLE AND SUBTITLE OCCUPANCT GRID MAP MERGING USING FEATURE MAPS			5a. CONTRACT NUMBER w56hzv-04-2-0001		
			5b. GRANT NUMBER		
			5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) James Overholt; Greg Hudas; Mohan Krishnan; Mark Paulik; Yasser Alnounou			5d. PROJECT NUMBER		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Advanced Mobility Laboratory, Department of Electrical and Computer Engineering, University of Detroit, Detroit, MI, 48221			8. PERFORMING ORGANIZATION REPORT NUMBER ; #21038		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army TARDEC, 6501 E. 11 Mile Rd, Warren, MI, 48397-5000			10. SPONSOR/MONITOR'S ACRONYM(S) TARDEC		
			11. SPONSOR/MONITOR'S REPORT NUMBER(S) #21038		
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES IASTED International Conference on Robotics and Applications, Cambridge, massachusetts					
14. ABSTRACT Multi-robot exploration and mapping studies have demonstrated that it is often more efficient to explore unknown areas in parallel rather than with a single agent. However, the question of how to integrate individual maps into a consistent global map remains an open research area. This problem, known as map merging, comprises the establishment of a frame of reference for multiple mobile robots, the identification of regions of map overlap, and the integration of individual maps to produce a global result. In this work, we build a hybrid map which integrates occupancy grid and feature data to solve this problem. This integrated representation permits fast and effective map merging. Experimental results are presented that demonstrate algorithm performance in a realistic scenario.					
15. SUBJECT TERMS multi-agent, multi-robot, map merging, hybrid maps, cooperative robotics, data fusion, robot sensing.					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Same as Report (SAR)	18. NUMBER OF PAGES 8	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

situations where the robots are not exploring large-scale areas. A modification of the previous work has been recently developed and uses an adaptive genetic algorithm to solve the optimization problem [7]. The authors use a similarity metric to adaptively change the crossover and mutation probabilities, making the algorithm converge faster. Although speed is improved, and a near-optimal transformation solution is obtained, the algorithm is still not very well-suited for real-time implementation with large-scale maps. Thus, even with heuristic techniques and efficient global optimization algorithms, searching the transformation space for occupancy grid maps is likely to be computationally prohibitive. Other relevant works have achieved better computational performance by extracting geometrical features from grid maps, comparing these features, and merging the maps if possible [4][8][9]. In [9], the authors compare distinctive features derived from patches of one map with entire secondary maps. The extracted features are recognized manually. In [8], it is assumed that each robot has its own map which consists of polylines (i.e., connected line segments). To merge two maps, the most salient polylines are extracted from each map. These lines are examined to identify polyline-correspondence hypotheses for the two maps. These are validated using a shape similarity metric and the transformation is applied if the match is deemed good. Another related technique, presented in [4], depends on line segments and the angles between them to identify, choose and validate possible transformations. The computation associated with this technique is proportional to the number of the line segments. However, this algorithm, depending solely on segment lines, was developed and tested for indoor environments and would require extension to accommodate the multiple irregular objects associated with unstructured outdoor mapping.

The work presented in this paper depends on the use of a new hybrid, feature-metric map, to address multi-robot map merging problems. It is similar in spirit to that presented in [4] and [8], in that it uses geometric features to identify suitable map transformations. However, the suitability of candidate transformations are evaluated using an occupancy-grid-based validation metric adapted from [6]. Thus a hybrid approach is introduced which combines a computationally efficient feature-based map transformation technique with a fine-grained grid-based validation measure. The paper presents two principle contributions. The first is a transformation identification algorithm which uses line-segments, circular arcs, and arbitrary curve features and a correlative, frequency-based, candidacy metric. The second is the computationally efficient integration of feature-based transformation candidate identification with an occupancy grid optimality measure. In this work, we are able to merge large-scale occupancy grid maps, collected from structured or unstructured environments, in a manner which permits real-time implementation.

This paper is organized as follows. Section II describes how we build a hybrid map. Then, in Section III we describe our map merging algorithm. The results of an experimental study are presented in Section IV whereas conclusions and suggestions for future work are provided in Section V.

II. Building Hybrid Maps

There are essentially three mapping paradigms used to represent environments: feature, topology, and occupancy grid. Each of these paradigms has advantages and drawbacks. Each kind could be more appropriate for specific tasks than others. For thorough surveys on map building, the reader is referred to [10],[3].

There can be many benefits to using hybrid maps, created by combining two or three of the paradigms. In such cases, one can exploit the advantages of the constituent paradigms and mitigate their limitations. For example, integrated occupancy grid and topology maps can be used for both local navigation (for which occupancy grid maps are superior) and path planning (which benefits from topological representations) when developing autonomous robots [11]. Moreover, it is relevant to mention that significant success has been achieved in solving SLAM problems when using hybrid maps [12], [13]. For this paper, hybridization produces a feature-metric map which is used for merging occupancy grid maps. The method used to construct the constituent maps is presented below.

A. Occupancy grid mapping

The robot's explored area is described by an occupancy grid map. This map is easy to build, describes the environment in detail, and is used for validation of the merging algorithm. It is also used to record the location of obstacles, which is necessary for safe local navigation. A grid map requires modest computational time to build, but can make large demands on memory to maintain. In most cases, grid mapping is done in a probabilistic fashion where each grid cell has a value between zero and one to represent the probability of being occupied by an obstacle. In our case, we set a threshold so as to decide whether each cell is occupied or not (1 or 0). Unexplored regions are assigned a probability of 0.5. Thus, the thresholded grid map represents each cell as being in one of three states, free, occupied, or unknown.

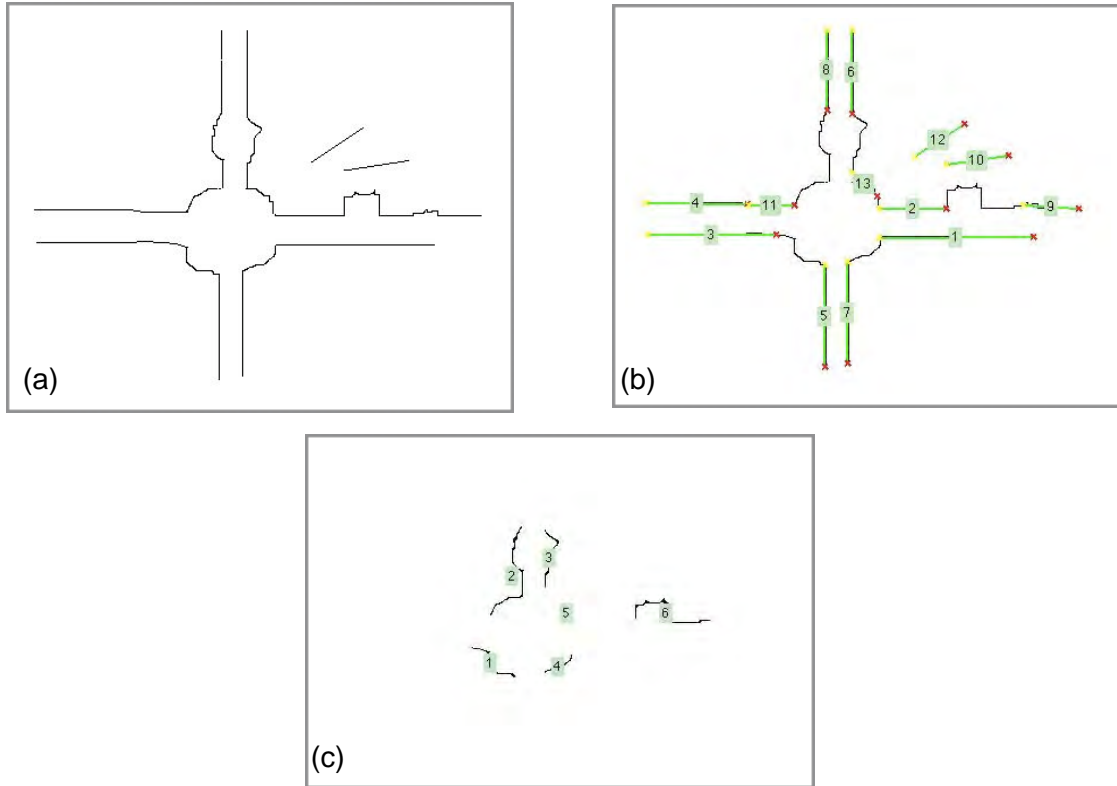


Figure 1. (a) A simulated environment, (b) Extracted lines, (c) The curves left after line extraction.

B. Feature mapping

We build our feature map on top of the occupancy grid. This kind of map represents the environment by parametric features. Our method is capable of representing a LIDAR scanned environment map in a parametric fashion. In general, indoor and outdoor environments can be described as a series of lines, circles, and arbitrary curves. We identify these features in a disjoint fashion by using a sequence of different techniques using a reduced map after each operation. First, we extract the line segments and circles from the grid map using the Hough transform.

The Hough transform has several characteristics that make it a good choice for extracting straight lines and circles (and other parametric curves) from a map. The transform is:

- Tolerant of gaps and immune to image noise, and
- Insensitive to partial deformation of the image features, e.g., it does not need all the pixels on one line to be contiguous or strictly collinear.

For each extracted line, we store the slope, length and the center of the line as parametric features. For the circles, we store the center and the radius. We then remove the map cells associated with the identified lines (designated as L1, L2, L3...) and circles (designated as C1, C2, C3...) from the map. The grid cells which remain represent arbitrary curves or sparse noise. We next parameterize these remain curves (designated as A1, A2, A3...) by measuring the average curve orientation and center of mass. It is relevant to point out, that a variety of shape descriptors could be used to represent the

Table 1. Parametric Features

Feature	Center X coord.	Center Y coord.	length / size	Orientation
L1	230	346	136	0
L2	205	307	59	0
L3	228	130	113	0
L4	200	117	90	0
L5	300	230	90	90
L6	84	254	74	90
L7	297	250	90	90
L8	82	232	71	90
L9	203	429	49	3
L10	162	364	55	171
L11	202	183	40	0
L12	144	331	52	146
L13	183	265	30	43
A1	241	205	104	42
A2	161	224	198	101
A3	144	258	111	98
A4	244	267	57	152
A5	196	275	38	72
A6	196	366	190	16

curves, including splines, polynomial expansions, etc. However, for this work a straightforward linear regression and center of mass description produced solid map merging results. Figure 1 depicts an environment generated in a simulated environment and the sequentially extracted features. Table 1 lists the associated features and their parameter values. It should be noted that lines or curves are considered to be features only if they are longer than a threshold value. This eliminates possibility of spurious map noise from being identified as genuine measurement data.

III. Map Merging Algorithm

The goal of our algorithm is to find the transformation (rotation and translation) matrix that is best for matching two maps. A series of candidate solutions are identified as map features are compared. To make a reject or accept decision, we adapt a map validation criteria presented in [6]. The metric is used to compare two occupancy grid maps after the application of a candidate transformation. First, we calculate an agreement and a disagreement function as follows.

$$agr(m1, m2) = \#\{p = (x, y) \mid m1[p] = m2[p] \in C\} \quad (1)$$

$$dis(m1, m2) = \#\{p = (x, y) \mid m1[p] \neq m2[p] \in C\} \quad (2)$$

where the agreement between two maps $m1$ and $m2$, is computed as the number of pixels p with associated coordinates (x, y) , such that point p in map-1 ($m1[p]$) is equal to point p in map-2 ($m2[p]$) (i.e., both *free*, or both *occupied*) where point p is a member of the set C comprising all points which have been identified as *free* or *occupied* in *both* maps. Similarly, the disagreement between two maps $m1$ and $m2$, is computed as the number of pixels p with associated coordinates (x, y) , such that point p in map-1 ($m1[p]$) is *not* equal to point p in map-2 ($m2[p]$) (i.e., one *free* and the other *occupied* and visa versa) where point p is a member of the set C comprising all points which have been identified as *free* or *occupied* in *both* maps.

These two functions, agr and dis , therefore represent how many pixels agree or disagree in the overlapping portions of the two maps respectively.

Second, we calculate a matching or validation factor as follows.

$$MF(m1, m2) = \frac{agr(m1, m2)}{agr(m1, m2) + dis(m1, m2)} \quad (3)$$

In typical cases, the match factor should approach 1.0. In our work, we set 0.85 as an acceptable threshold for validating the match between two maps. In reality, one cannot easily lower this factor because wrong merge decisions can produce useless global maps and erroneous robot poses. At the same time, a lower factor might be suitable when trying to merge low accuracy maps.

From the extracted features, we need to look for matches between the two sets of features. First, for each line L or curve A from the first map, we define a corresponding set which have lines $\{L_i, L_{i+1}, \dots\}$ or curves $\{A_i, A_{i+1}, \dots\}$ from the other map, which are approximately the same size of that line or curve, respectively. It is useful to establish a tolerance to account for uncertainty in the map building process (we accepted features that were within 15% of each other). Second, for each feature in the first map, we determine the absolute angle difference between it and the set of corresponding (i.e., similar length) features in the second map. This produces multiple sets of candidate rotation angles (one set for each feature in the first map). This done, we sort the candidate rotation angles, based on frequency of occurrence across all the sets, in decreasing order (i.e., the most frequent rotation angle first) after quantizing them to a 1° resolution. From this sorted set, we use the first angle as a possible solution for rotation. Next, we follow a similar procedure to identify the translation candidates. We begin by rotating the second map by the candidate angle, and then, for each feature's center of mass coordinates (this time including circular features $C1, C2, \dots$), subtract the center of mass coordinates associated with each member in its corresponding set (this set comprising like features -- line to line, circle to circle, etc. -- in the second map with similar size). Each feature thus produces a set of signed coordinate differences (tuples), which are then sorted based on frequency of occurrence in decreasing order. These candidate translations are then applied sequentially to the second (rotated) map, and the corresponding matching/validation factors calculated to score the quality of the transformations.

The algorithm then continues this process, using the next candidate rotation and the corresponding translations. Ultimately, we are seeking the largest matching factor across all candidate rotations. In the end, if the best matching factor exceeds the decision threshold (0.85 for example), the corresponding transformation is used to merge the maps; otherwise the maps are not merged. The pseudocode below describes our algorithm more succinctly.

Algorithm 1: Map Merging

```
1. // Build feature maps
2. F1 = Extract feature (map 1)
3. F2 = Extract feature (map 2)
4. Calculate the rotation candidates R1, R2, ..., Rn.
5. sort the candidate rotation array
6. for k = 1, ..., n
7.   rotate(map 1, map 2, Rk)
8.   calculate the translation candidates T1, T2, ..., Tm
9.   sort the candidate translation array
10.  for j = 1, ... , m
11.    Translate (map 1, map 2, Tj)
12.    match factor = Validate (map 1, map 2 )
13.    if (match factor > current factor)
14.      current factor = merge factor
15.      save current transformation TR (Tj, Rk)
16.    end if
17.  end for
18. end for
19. if current factor > 0.85
20.   global map = merge(map 1, map 2 , TR)
21. end
```



Figure 2. Our two robots: RA on the right, RB on the left.

IV. Experimental Results

An experimental study was undertaken to validate the algorithms presented in this paper. Two Departmental robots, “RA” and “RB” were used to collect mapping data outdoors on site. The area explored bordered a pedestrian mall, a building and some decorative cul-de-sacs. The robots were built onsite and are depicted in Figure 2. For these experiments, each robot was equipped with LIDAR and an odometry system based on the robot’s kinematics and motor encoders. An open source server/client and simulation system was used for software development and implementation.

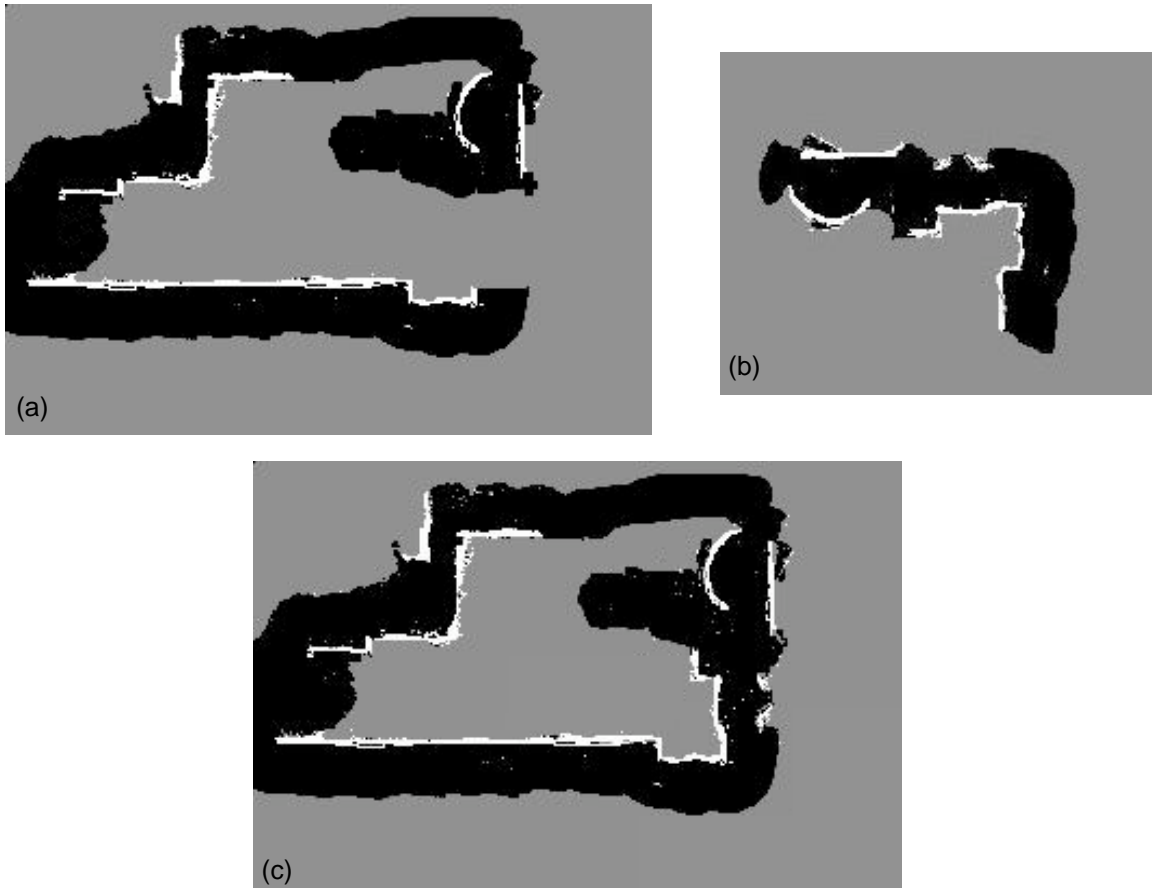


Figure 3. Two maps, built independently, have common area (a) Built by RA (b) built by RB. The two maps (c) The global map

The two robots independently used probabilistic grid mapping based on the Rao-Blackwellized particle filter approach explained in [14]. The filter undertakes state estimation of the robot pose and map based on a motion model and a measurement model. The resulting probabilistic map is thresholded to identify three types of regions: occupied, free, unknown. Occupied grids have a probability of 0.7 or more while free cells have a probability 0.3 or less and the unknown grids are in between (> 0.3 & < 0.7). The resolution of our map was set to 20 cm/pixel. The size of our environment for this experiment was 200*300 pixels. We intentionally let the two robots pass by a small common area and then ran the algorithm to see how consistently it could merge the maps to produce a global environment map. Figure 3 shows the two separate maps and the final integrated global map.

Our algorithm easily identified the common area and returned the transformation matrix within milliseconds (with interpretive coding implementations, e.g., Octave, Scilab etc.). The algorithm computational complexity is linearly proportional to the number of the features in a map. Typical values for the agreement and disagreement functions were, $agr = 4392$, $dis = 139$ which leads to a 96% matching factor. Once the global map is obtained it is sent to the other robot to synchronize systems. It is relevant to point out, that when the global map is formed, grid-cells which conflict, i.e., points where one map indicates the cell is *free* and the other map indicates *occupied*, are returned to the unknown state. Also, this work easily extends to multi-agent systems with more than two robots by executing the algorithm in a pairwise fashion.

The map matching algorithm performance for maps similar to that of Figure 3 is consistently good. When the region mapped is dominated by arbitrary curves and contains few clear line segments, the map matching algorithm performance is reduced. This is due to the method used to identify the curve orientation, which is sensitive to variations in the portion of the curved area mapped and the presence of noise. This can be addressed with the use of more sophisticated curve parameterizations (multi-resolution and disjoint curve parsing present interesting possibilities).

Finally, it is illustrative to point out that this work can also be used to solve localization problems. Of particular interest are situations where a robot has a discontinuity in its pose estimation process (e.g., the kidnapped robot problem), or where a robot has been given a map yet does not know where it is in this map (e.g. global localization). In such cases, the robot can explore the area until it finds a match between its current map and a previous or given map. Also the algorithm presented in this paper can potentially be applied to the loop detection problem wherein previously visited areas need to be identified.

V. Conclusions and Future Work

In this paper we introduced a hybrid, feature-metric, map for occupancy grid map merging. Our method has produces good results with semi-unstructured outdoor environments, and has demonstrated impressive algorithm speed. The experimental results indicate promising potential for the use of hybrid feature-metric maps for merging multi-robot maps.

Future work will likely proceed along a number of avenues. Arbitrary curves need to be parsed and/or parameterized in a more robust fashion to permit effective application to fully unstructured outdoor environments. Furthermore, the proposed merging algorithm could be integrated with probabilistic scan matching steps to improve fine resolution alignment of map segments. The current method depends to a degree on the quality or accuracy of the constituent maps being merged; therefore, the introduction of a probabilistic framework for feature parameterization could permit a more measured degradation in performance as map noise increases.

VI. REFERENCES

- [1] B. Yamauchi, "Frontier-based exploration using multiple robots," *Proceeding of the 2nd International Conf. on Autonomous Agents*, Minneapolis, MN, 1998, 47–53.
- [2] A. Howard, L. E. Parker, and G. S. Sukhatme, "The SDR experience: Experiments with a large-scale heterogenous mobile robot team," *Proceedings of the 9th International Symposium on Experimental Robotics*, Singapore, June 2004.
- [3] S. Thrun, "Robotic mapping: a survey," *Exploring artificial intelligence in the new millennium*, Morgan Kaufmann Publishers Inc, San Francisco, CA, 2003.
- [4] F. Amigoni, S. Gasparini, and M. Gini, "Building segment-based maps without pose information," *Proceeding of IEEE*, 94(7), 2006, 1340-1359.
- [5] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, B. Stewart, "Distributed multi-robot exploration and mapping," *Proceedings of the IEEE*, (94)7, 2006, 1325-1339.
- [6] A. Birk, and S. Carpin, "Merging occupancy grid maps from multiple robots," *Proceedings of the IEEE*, 94(7), 2006, 1384-1397.
- [7] Xin Ma, Rui Guo, Yibin Li, and Weidong Chen, "Adaptive genetic algorithm for occupancy grid maps merging," in *7th World Congress on Intelligent Control and Automation, WCICA*, Chongqing, 2008, 5716-5720.
- [8] R. Lakaemper, L. J. Latecki, and D. Wolter, "Incremental multi-robot mapping," *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, Edmonton, Canada, August 2005.
- [9] K. Konolige, D. Fox, B. Limketkai, J. Ko, and B. Stewart, "Map merging for distributed robot navigation," *Proceedings of IEEE/RSJ International Conf. on Intelligent Robots and Systems (IROS)*, Menlo Park, CA, 2003, 212-217.

- [10] Chen Chen, and Yinhang Cheng, "Research on map building by mobile robots," *Second International Symposium on Intelligent Information Technology Application IITA*, Shanghai 2008, 673-677.
- [11] A. Bandera, C. Urdiales, and F. Sandoval, "An hierarchical approach to grid-based and topological maps integration for autonomous indoor navigation," *Proceedings of IEEE/RSJ International Conf. on Intelligent Robotics and Systems*, Maui, HI, 2001, 883-888.
- [12] F. Ferreira, I. Amorim, R. Rocha, and J. Dias, "T-SLAM: Registering topological and geometric maps for robot localization in large environments," *IEEE International Conf. on Multisensor Fusion and Integration for Intelligent Systems, MFI*, Seoul, 2008, 392-398.
- [13] J.-L. Blanco, and J.-A. Fernandez-Madrigal, J. Gonzalez, "Toward a unified bayesian approach to hybrid metric-topological SLAM," *IEEE Transactions on Robotics*, 24(2), April 2008, 259-270.
- [14] G. Riseti, C. Stachniss, and W. Burgard, "Improved Techniques for grid mapping with Rao-Blackwellized Particle Filter," *IEEE Transactions on Robotics*, 23(1), February 2007, 34-46.